LAPORAN TUGAS BESAR  
IF3270 PEMBELAJARAN MESIN

ARTIFICIAL NEURAL NETWORK  
BAGIAN A:   
IMPLEMENTASI FORWARD PROPAGATION UNTUK FEED FORWARD NEURAL NETWORK

****

Oleh:

Eugene Yap Jin Quan 13521074

Michael Utama 13521137

Johann Christian Kandani 13521138

Dewana Gustavus Haraka Otang 13521173

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2024**

# DAFTAR ISI

[**DAFTAR ISI 2**](#_3u3aj3gc6mfq)

[**I: SPESIFIKASI 3**](#_pp7ipu4efcj6)

[**II: IMPLEMENTASI 4**](#_jyxpea7vd5f0)

[Deskripsi Implementasi 4](#_3acx5qxtwhtl)

[Perbandingan Algoritma Dengan Perhitungan Manual 6](#_gkkcewi11zcj)

[**III: PENGUJIAN 8**](#_duxwm315kzoj)

[Hasil Pengujian 8](#_1zhe17p2mouc)

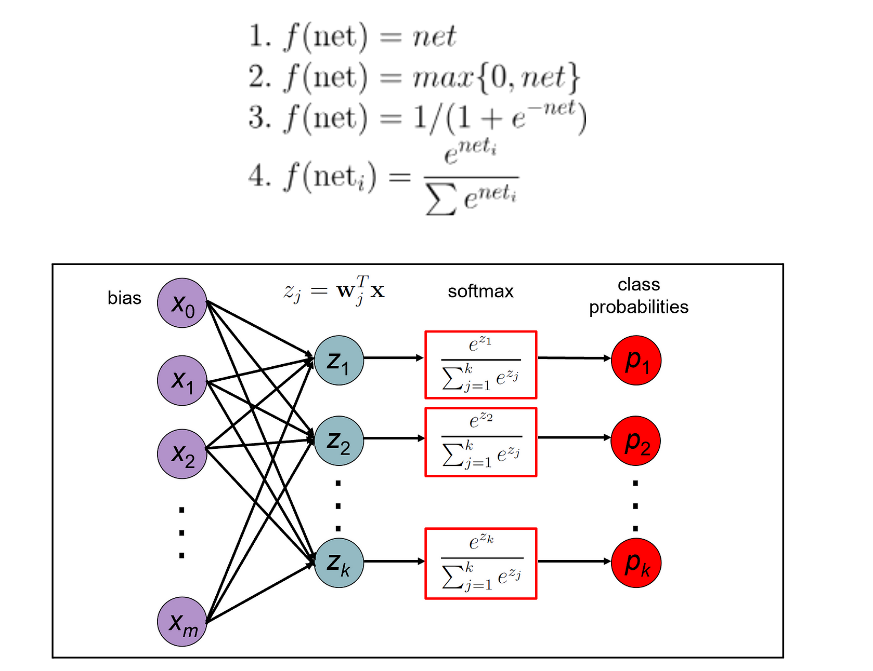
[**IV: PEMBAGIAN TUGAS 10**](#_3zsmy68lzk2b)

[**LAMPIRAN 11**](#_g6p8y0hxxcol)

# I: SPESIFIKASI

Pada Tugas Besar Bagian A, kami diinstruksikan untuk merancang jaringan saraf tiruan pada bagian feed forward neural network (FFNN) menggunakan bahasa Python. Berikut adalah detail spesifikasi.

1. Program berfungsi sebagai neural network yang dapat melakukan feed forward dari input yang diberikan.
2. Program dapat menerima masukan dari file JSON.
3. Implementasi pada fungsi aktivasi berikut:
   1. Linear
   2. ReLU
   3. Sigmoid
   4. Softmax



*Fungsi aktivasi dan ilustrasi fungsi softmax*

1. Program dapat menyimpan bobot (weights) dan struktur model.
2. Implementasi forward propagation untuk FFNN dengan kemampuan:
   1. Menampilkan model berupa struktur jaringan dan bobotnya, formatnya bebas.
   2. Memberikan output untuk input 1 instance.
   3. Memberikan output untuk input berupa batch.
3. Direkomendasikan membuat layer dan fungsi aktivasi secara modular.

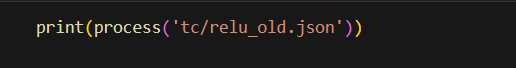
Batasan dari implementasi adalah satu layer memiliki fungsi aktivasi yang sama; layer yang berbeda dapat memiliki fungsi aktivasi yang berbeda.

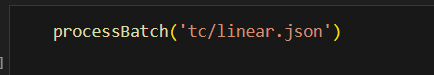
# II: IMPLEMENTASI

## Deskripsi Implementasi

Berikut adalah detail implementasi model FFNN tugas besar ini.

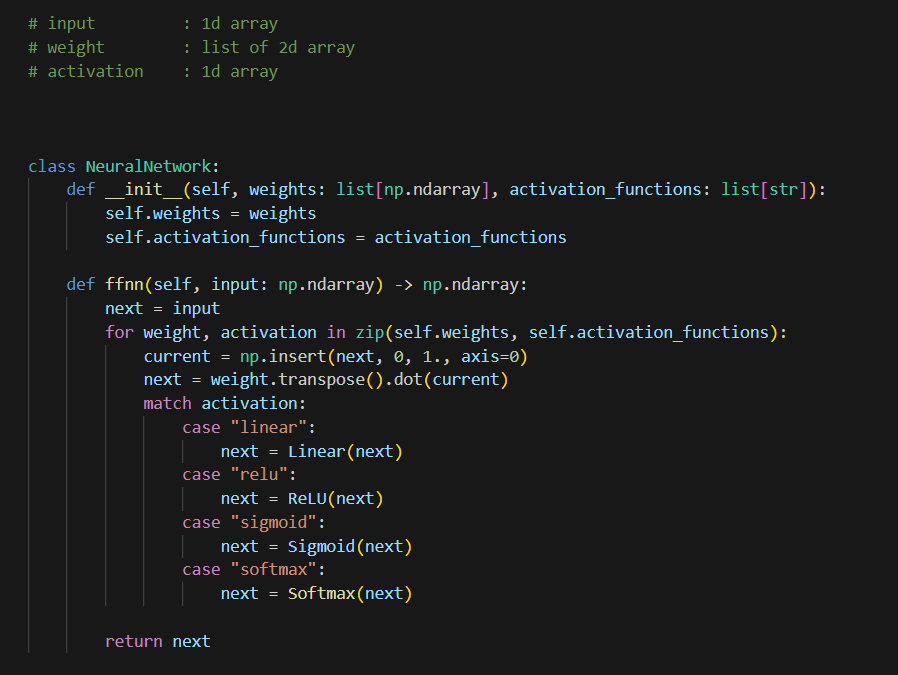
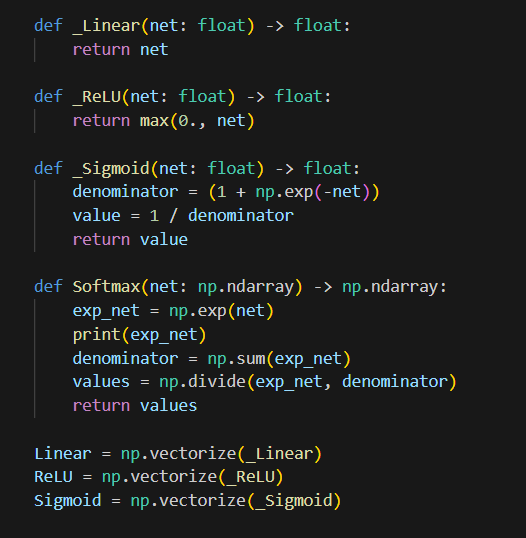
* Program melakukan input berdasarkan nama *file* yang dimasukkan pada fungsi integrasi yang digunakan untuk menghitung hasil akhir.





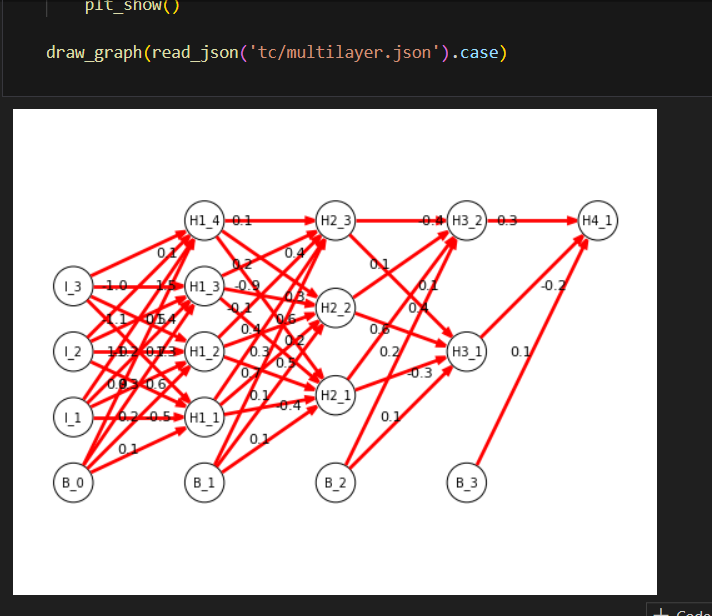
*Masukan berupa path dari file kasus uji*

* Fungsi aktivasi dibuat sebagai fungsi yang mengambil *numpy array* dan mengembalikan *numpy array* (dengan atau tanpa bantuan *np.vectorize*). Fungsi-fungsi aktivasi tersebut diimplementasikan secara terpisah dari fungsi FFNN.

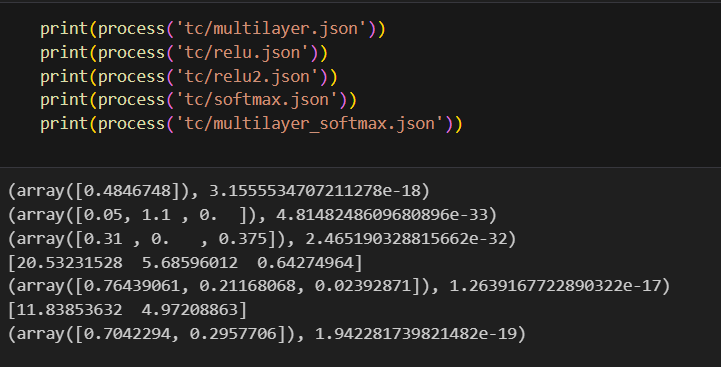


*Fungsi aktivasi diimplementasikan secara modular*

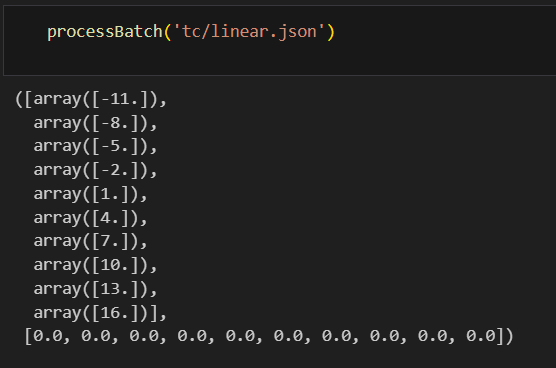
* Fungsi FFNN meminta input, bobot, dan fungsi aktivasi. Untuk tiap layer, akan dilakukan perkalian matriks dan vektor antara bobot (*transposed*) dan nilai layer sebelumnya ditambah bias.
* Seluruh fungsi yang telah diimplementasikan diintegrasikan ke dalam tiga fungsi keluaran. Ketiga fungsi keluaran masing-masing menghasilkan keluaran berupa visualisasi model, hasil pemrosesan satu *instance*, dan hasil pemrosesan *batch*.



*Visualisasi model FFNN*



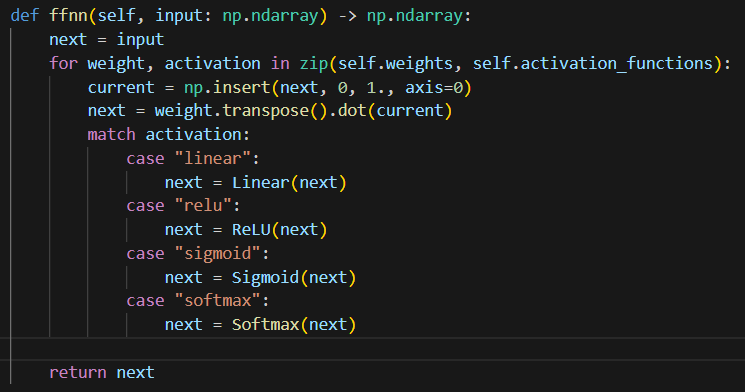
*Contoh keluaran pemrosesan 1 instance (implementasi menggunakan fungsi process)*

**

*Contoh keluaran pemrosesan batch (implementasi menggunakan fungsi processBatch)*

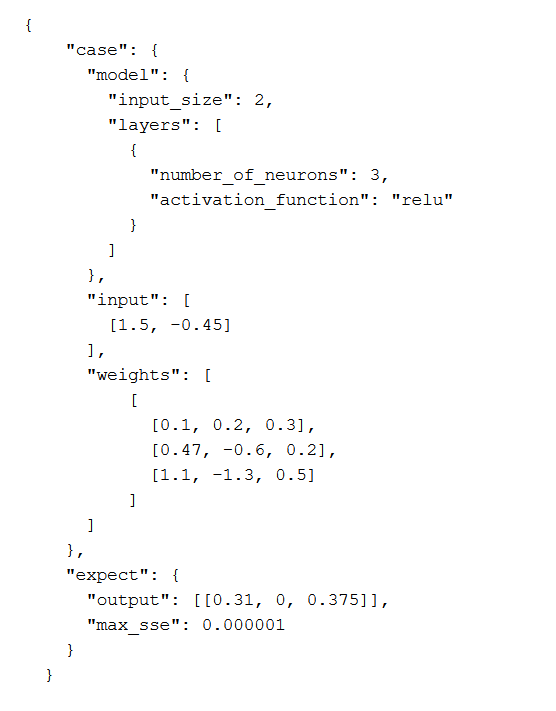
## Perbandingan Algoritma Dengan Perhitungan Manual

Algoritma FFNN yang diimplementasikan merupakan rumus perkalian *dot* antara vektor bobot-bobot input dalam sebuah neuron dengan vektor input. Keluaran dari FFNN merupakan nilai keluaran dari sebuah neuron. Apabila sebuah *layer* terdiri atas beberapa neuron output, hasil perhitungan output dari *layer* tersebut akan menghasilkan sebuah vektor yang memiliki komponen sebanyak jumlah neuron.



*Implementasi algoritma FFNN untuk satu neuron*

Sebelum program melakukan perhitungan pada setiap *layer* FFNN, program akan melakukan pembacaan input berupa file JSON. File masukan JSON berisi informasi jumlah input, vektor nilai input, *array* berisi informasi setiap *layer*, masing-masing berisi fungsi aktivasi *layer*, jumlah neuron, dan daftar bobot input untuk setiap neuron dalam *layer*

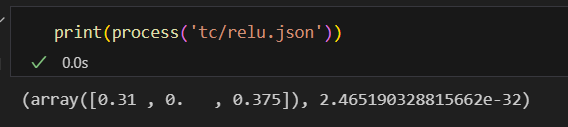


*relu.json*

Sebagai ilustrasi, apabila masukan berupa kasus uji relu.json, informasi yang dibaca program adalah sebagai berikut.

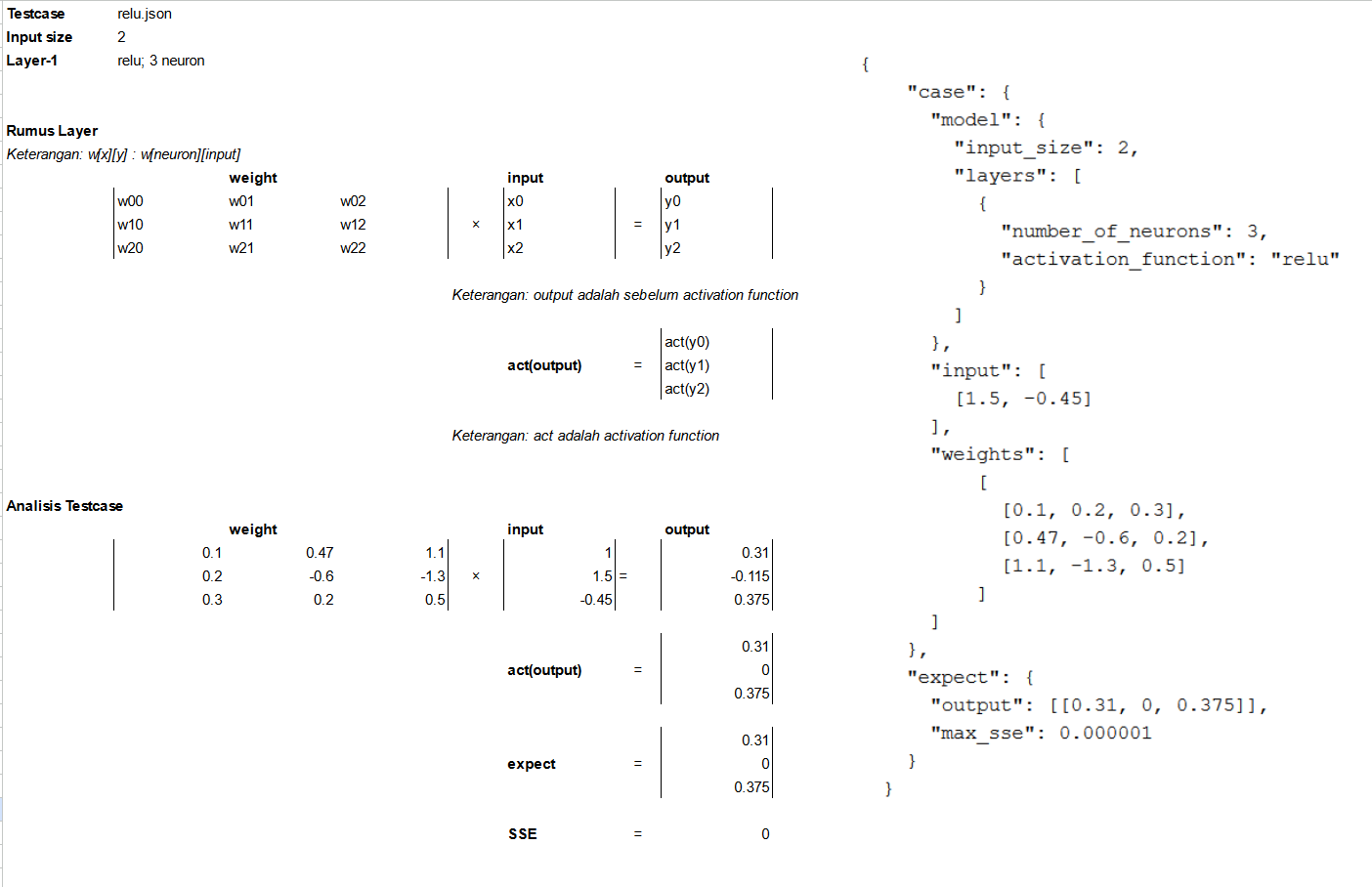
* Vektor input = [1.5, -0.45]
* Jumlah *layer* = 1
* *Layer-0* memiliki 3 neuron dan fungsi aktivasi ReLU
  + Vektor bobot input untuk input 0 (bias) = [0.1, 0.2, 0.3]
  + Vektor bobot input untuk input 1 (x1) = [0.47, -0.6, 0.2]
  + Vektor bobot input untuk input 2 (x2) = [1.1, -1.3, 0.5]
  + Catatan, untuk setiap input, vektor bobot input adalah [*bobot\_neuron0, bobot\_neuron1, bobot\_neuron2, …*]

Dengan menggunakan algoritma yang telah diimplementasi, fungsi FFNN akan dipanggil sebanyak jumlah neuron pada sebuah *layer* untuk menghasilkan vektor nilai output. Untuk kasus uji relu.json, perhitungan menggunakan program mengeluarkan hasil berikut.



*Hasil perhitungan relu.json menggunakan program*

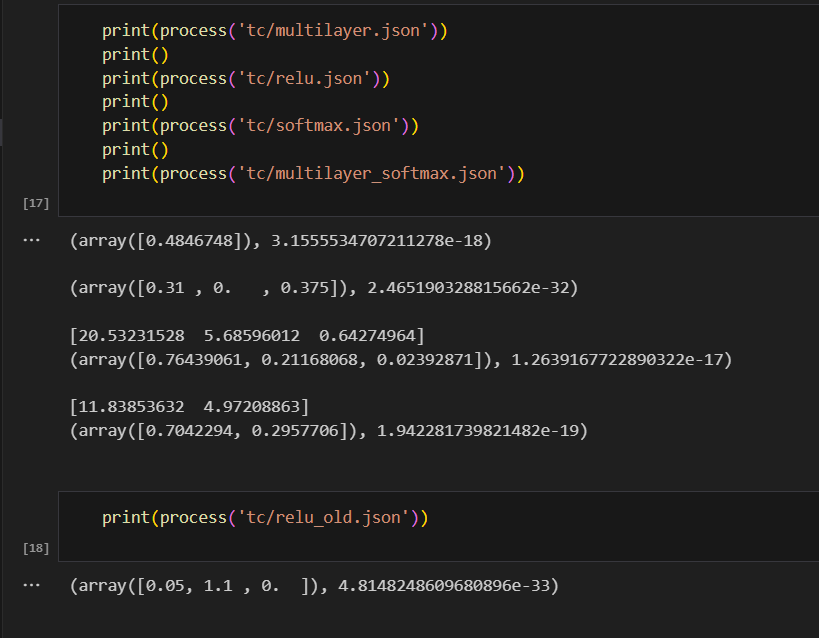
Melalui observasi, operasi perkalian *dot* antara vektor input dan vektor bobot yang dilakukan pada setiap neuron menyerupai operasi perkalian matriks yang digunakan untuk menghitung output sebuah layer secara manual.



*Perhitungan manual untuk kasus uji relu.json*

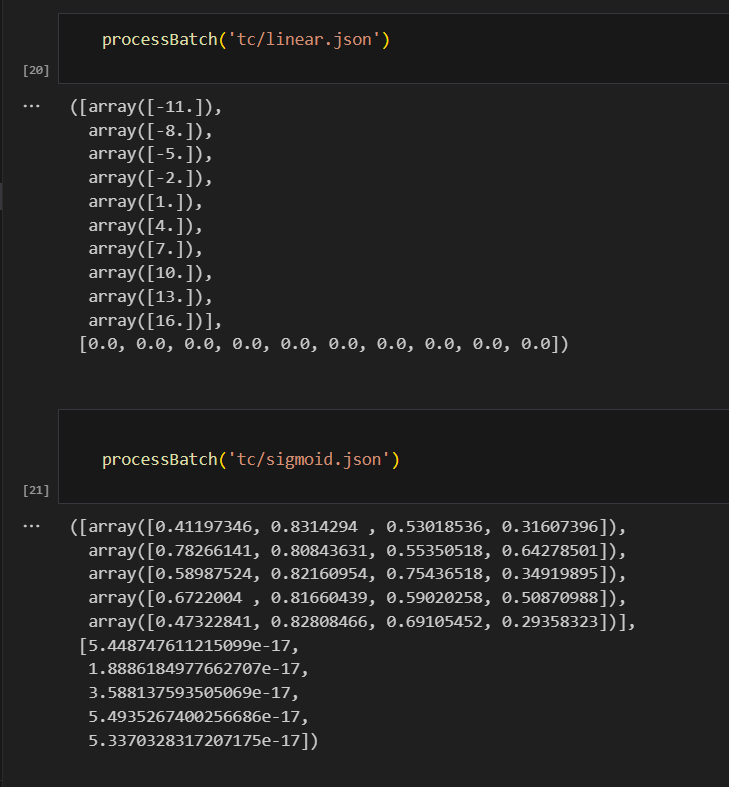
# III: PENGUJIAN

## Hasil Pengujian



*Hasil pengujian untuk pemrosesan single instance FFNN*

Gambar di atas adalah hasil pengujian satu instance, untuk beberapa testcase. Nilai *array* di kiri menunjukkan nilai output layer, sedangkan nilai di kanan menunjukkan *sum of squared error*.



*Hasil pengujian untuk pemrosesan batch FFNN*

Gambar di atas adalah hasil pengujian batch. *Array* pertama menunjukkan list hasil (input pertama menghasilkan output -11, dst). *Array* kedua menunjukkan nilai *sum of squared error* tiap prosesnya.

# IV: PEMBAGIAN TUGAS

Pembagian tugas pada pengerjaan Tugas Besar IF3270 Bagian A adalah sebagai berikut.

| NIM | Nama | Tugas |
| --- | --- | --- |
| 13521074 | Eugene Yap Jin Quan | Integrasi, visualisasi |
| 13521137 | Michael Utama | FFNN, integrasi |
| 13521138 | Johann Christian Kandani | Visualisasi, baca JSON |
| 13521173 | Dewana Gustavus Haraka Otang | Fungsi aktivasi, baca JSON, visualisasi |

# LAMPIRAN

Dokumen Spesifikasi: [IF3270 - Spesifikasi Tugas Besar](https://docs.google.com/document/d/1DO4g3y76CX5uN3yAMZ7KcvbKmRnQp2ZpdIpls1Ik4wI/edit?usp=sharing)

Kasus Pengujian: [Test Case Bagian A](https://drive.google.com/drive/folders/1xeafmjch-D-zm1ilOFVSdDlC0jgrrjmU)

Kasus Pengujian (versi 0): [relu.json](https://drive.google.com/file/d/16lDA6ar_p_nA68cTBFO_bKTAi1s9oAHA/view)

Perhitungan Manual: [Perhitungan Manual Bagian A - Kelompok 26](https://docs.google.com/spreadsheets/d/1qpQGtud0B11qWmNt0Z4Js8Usye5bjZpCJqR_KiTJBHQ/edit?usp=sharing)